

## Linked list

A non-empty zero-indexed array A consisting of N integers is given.

Array A represents a linked list. A list is constructed from this array as follows:

- the first node (the head) is located at index 0
- the value of a node located at index K is A[K]
- if the value of the node is -1 then it is the last node of the list
- otherwise, the successor of a node located at index K is located at index A[K] (you can assume that A[K] is a valid index, that is  $0 \leq A[K] < N$ ).

For example, for array A such that:

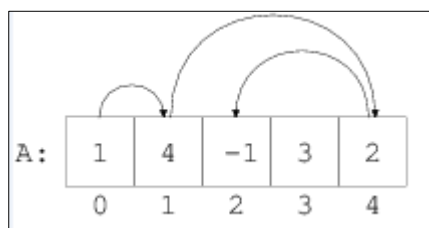
A[0] = 1

A[1] = 4

A[2] = -1

A[3] = 3

A[4] = 2



The following list is constructed:

- the first node (the head) is located at index 0 and has a value of 1
- the second node is located at index 1 and has a value of 4
- the third node is located at index 4 and has a value of 2
- the fourth node is located at index 2 and has a value of -1.

Write a function:

```
object Solution { def solution(a: Array[Int]): Int }
```

that, given a non-empty zero-indexed array A consisting of N integers, returns the length of the list constructed from A in the above manner.

For example, given array A such that:

A[0] = 1

A[1] = 4

A[2] = -1

A[3] = 3

A[4] = 2

the function should return 4, as explained in the example above.

Assume that:

- $N$  is an integer within the range  $[1..200,000]$
- each element of array  $A$  is an integer within the range  $[-1..N-1]$
- it will always be possible to construct the list and its length will be finite

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.